IN THE CLAIMS

Applicants note that claims 4 and 20 are not amended to overcome prior art but to correct typographical errors. The amendments made to claims 4 and 20 are not narrowing in scope and therefore no prosecution history estoppel arises from the amendments to claims 4 and 20. *Festo Corp v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 62 U.S.P.Q.2d 1705, 1711-1712 (2002); 56 U.S.P.Q.2d 1865, 1870 (Fed. Cir. 2000). Further, the amendments made to claims 4 and 20 were not made for a substantial reason related to patentability and therefore no prosecution history estoppel arises from such amendments. *See Festo Corp.*, 62 U.S.P.Q.2d 1705 at 1707 (2002); *Warner-Jenkinson Co. v. Hilton Davis Chemical Co.*, 41 U.S.P.Q.2d 1865, 1873 (U.S. 1997).

1   Claim 1 (original): A timer management system for managing timers in both a
2   synchronous and asynchronous system comprising:
3   an application program interface (API) providing a set of synchronous
4   functions allowing an application to functionally operate a timer;
5   a timer database for storing timer-related information; and
6   a timer services detecting the expiring of said timer, wherein a handle function of said
7   timer services allows said application to act on an expired timer without incurring an
8   illegal time-out message.

1   Claim 2 (original): The timer management system as recited in claim 1, wherein said
2   application performs the following operations on said timer via said API:
3   creating said timer from an allocated block of system memory;
4   activating said timer; and
5   reinitializing said timer using said allocated block of system memory.

1   Claim 3 (original): The timer management system as recited in claim 1, wherein said
2   application performs the following operation on said timer via said API:
3   creating said timer from an allocated block of system memory; and
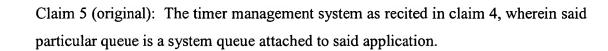
4   activating said timer;

5   ,wherein said timer expires and said timer services sends synchronously a

6   time-out message to said application, wherein said time-out message is sent using

7   said allocated block of system memory.

1   Claim 4 (currently amended): The timer management system as recited in claim 1,

2   wherein said application performs the following operation on said timer via said API:

3   creating said timer from an allocated block of system memory; and

4   activating said timer;

5   wherein said timer expires and said timer services sends a time-out message to

6   a particular queue, wherein said timer is in an expired state in an asynchronous state

7   machine.

1   Claim 5 (original): The timer management system as recited in claim 4, wherein said

2   particular queue is a system queue attached to said application.

1   Claim 6 (original): The timer management system as recited in claim 4, wherein said

2   application receives said time-out message, wherein said handle function transfers

3   said timer from said expired state in said asynchronous state machine to an idle state

4   in a synchronous state machine, wherein said handle function allows said application

5   to synchronously act on said timer.

1   Claim 7 (original): The timer management system as recited in claim 4, wherein said

2   application stops said timer, wherein said timer is in an idle state in said asynchronous

3   state machine with said time-out message being queued.

1   Claim 8 (original): The timer management system as recited in claim 7, wherein said

2   time-out message is dequeued, wherein said handle function transfers said timer from

3   said idle state in said asynchronous state machine to an idle state in a synchronous

4   state machine, wherein said handle function allows said application to synchronously

5   ,   act on said timer.

1　Claim 9 (original): The timer management system as recited in claim 7, wherein said
2　timer is deleted by said application, wherein said timer is in a state in said
3　asynchronous state machine in which said timer is deleted and said time-out message
4　is queued, wherein said time-out message is dequeued, wherein said handle function
5　transfers said timer from said state in said asynchronous state machine in which said
6　timer is deleted and said time-out message is queued to a non-existent state in a
7　synchronous state machine, wherein said handle function allows said application to
8　synchronously act on said timer.

1　Claim 10 (original): The timer management system as recited in claim 7, wherein
2　said timer is activated by said application, wherein said timer is in a running state in
3　said asynchronous state machine with said time-out message being queued.

1　Claim 11 (original): The timer management system as recited in claim 10, wherein
2　said timer is deleted by said application, wherein said timer is in a state in said
3　asynchronous state machine in which said timer is deleted and said time-out message
4　is queued, wherein said time-out message is dequeued, wherein said handle function
5　transfers said timer from said state in said asynchronous state machine in which said
6　timer is deleted and said time-out message is queued to a non-existent state in a
7　synchronous state machine, wherein said handle function allows said application to
8　synchronously act on said timer.

1　Claim 12 (original): The timer management system as recited in claim 10, wherein
2　said timer is stopped by said application, wherein said timer is in said idle state in
3　said asynchronous state machine with said time-out message being queued, wherein
4　said time-out message is dequeued, wherein said handle function transfers said timer
5　from said idle state in said asynchronous state machine to an idle state in a
6　synchronous state machine, wherein said handle function allows said application to
7　synchronously act on said timer.

1   Claim 13 (original): The timer management system as recited in claim 10, wherein

2   said time-out message is dequeued, wherein said handle function transfers said timer

3   from said running state in said asynchronous state machine to a running state in a

4   synchronous state machine, wherein said handle function allows said application to

5   synchronously act on said timer.

1   Claim 14 (original): The timer management system as recited in claim 4, wherein

2   said application deletes said timer, wherein said timer is in a state in said

3   asynchronous state machine in which said timer is deleted and said time-out message

4   is queued, wherein said time-out message is dequeued, wherein handle function

5   transfers said timer from said state in said asynchronous state machine in which said

6   timer is deleted and said time-out message is queued to a non-existent state in a

7   synchronous state machine, wherein said handle function allows said application to

8   synchronously act on said timer.

1   Claim 15 (original): The timer management system as recited in claim 4, wherein

2   said application activates said timer, wherein said timer is in a running state in said

3   asynchronous state machine with said time-out message being queued.

1   Claim 16 (original): The timer management system as recited in claim 15, wherein

2   said timer is deleted by said application, wherein said timer is in a state in said

3   asynchronous state machine in which said timer is. deleted and said time-out message

4   is queued, wherein said time-out message is dequeued, wherein said handle function

5   transfers said timer from said state in said asynchronous state machine in which said

6   timer is deleted and said time-out message is queued to a non-existent state in a

7   synchronous state machine, wherein said handle function allows said application to

8   synchronously act on said timer.

1   Claim 17 (original): The timer management system as recited in claim 15, wherein

2   said timer is stopped by said application, wherein said timer is in an idle state in said

3   asynchronous state machine with said time-out message being queued, wherein said

4   time-out message is dequeued, wherein said handle function transfers said timer from

5   said idle state in said asynchronous state machine to an idle state in a synchronous

6   state machine, wherein said handle function allows said application to synchronously

7   act on said timer.

1   Claim 18 (original): The timer management system as recited in claim 15, wherein

2   said time-out message is dequeued, wherein said handle function transfers said timer

3   from said running state in said asynchronous state machine to a running state in a

4   synchronous state machine, wherein said handle function allows said application to

5   synchronously act on said timer.

1   Claim 19 (original): The timer management system as recited in claim 1, wherein

2   said API is a DLL file.

1   Claim 20 (currently amended): A method for managing timers in both a synchronous

2   and asynchronous system comprising the steps of:

3   creating a timer from an allocated block of system memory by an application

4   via an application program interface (API);

5   activating said timer;

6   expiring of said timer; and

7   sending a time-out message to a particular queue when said timer expires,

8   wherein said timer is in an expired state in an asynchronous state machine, wherein a

9   handle function allows said application to act on said expired timer without incurring

10   an illegal time-out message.

1   Claim 21 (original): The method as recited in claim 20, wherein said particular queue

2   is a system queue attached to said application.

1   Claim 22 (original): The method as recited in claim 20 further comprising the step

2   of:

3   receiving said time-out message by said application, wherein said handle

4   function transfers said timer from said expired state in said asynchronous state

5   machine to an idle state in a synchronous state machine, wherein said handle function

6   allows said application to synchronously act on said timer.

1   Claim 23 (original): The method as recited in claim 20 further comprising the step

2   of:

3   stopping said timer by said application, wherein said timer is in an idle state in

4   said asynchronous state machine with said time-out message being queued.

1   Claim 24 (original): The method as recited in claim 23, wherein said time-out

2   message is dequeued, wherein said handle function transfers said timer from said idle

3   state in said asynchronous state machine to an idle state in a synchronous state

4   machine, wherein said handle function allows said application to synchronously act

5   on said timer.

1   Claim 25 (original): The method as recited in claim 23 further comprising the step

2   of:

3   deleting said timer by said application, wherein said timer is in a state in said

4   asynchronous state machine in which said timer is deleted and said time-out message

5   is queued, wherein said time-out message is dequeued, wherein said handle function

6   transfers said timer from said state in said asynchronous state machine in which said

7   timer is deleted and said time-out message is queued to a non-existent state in a

8   synchronous state machine, wherein said handle function allows said application to

9   synchronously act on said timer.

1   Claim 26 (original): The method as recited in claim 23 further comprising the step

2   of:

3   activating said timer by said application, wherein said timer is in a running

4   state in said asynchronous state machine with said time-out message being queued.

12

1 Claim 27 (original): The method as recited in claim 26 further comprising the step

2 of:

3   deleting said timer by said application, wherein said timer is in a state in said

4 asynchronous state machine in which said timer is deleted and said time-out message

5 is queued, wherein said time-out message is dequeued, wherein said handle function

6 transfers said timer from said state in said asynchronous state machine in which said

7 timer is deleted and said time-out message is queued to a non-existent state in a

8 synchronous state machine, wherein said handle function allows said application to

9 synchronously act on said timer.

1 Claim 28 (original): The method as recited in claim 26 further comprising the step

2 of:

3   stopping said timer by said application, wherein said timer is in said idle state

4 in said asynchronous state machine with said time-out message being queued,

5 wherein said time-out message is dequeued, wherein said handle function transfers

6 said timer from said idle state in said asynchronous state machine to an idle state in a

7 synchronous state machine, wherein said handle function allows said application to

8 synchronously act on said timer.

1 Claim 29 (original): The method as recited in claim 26, wherein said time-out

2 message is dequeued, wherein said handle function transfers said timer from said

3 running state in said asynchronous state machine to a running state in a synchronous

4 state machine, wherein said handle function allows said application to synchronously

5 act on said timer.

1 Claim 30 (original): The method as recited in claim 20 further comprising the step

2 of:

3   deleting said timer by said application, wherein said timer is in a state in said

4 asynchronous state machine in which said timer is deleted and said time-out message

5 is queued, wherein said time-out message is dequeued, wherein said handle function

6   transfers said timer from said state in said asynchronous state machine in which said

7   timer is deleted and said time-out message is queued to a non-existent state in a

8   synchronous state machine, wherein said handle function allows said application to

9   synchronously act on said timer.

1   Claim 31 (original): The method as recited in claim 20 further comprising the step

2   of:

3   activating said timer by said application, wherein said timer is in a running

4   state in said asynchronous state machine with said time-out message being queued.

1   Claim 32 (original): The method as recited in claim 31 further comprising the step

2   of:

3   deleting said timer by said application, wherein said timer is in a state in said

4   asynchronous state machine in which said timer is deleted and said time-out message

5   is queued, wherein said time-out message is dequeued, wherein said handle function

6   transfers said timer from said state in said asynchronous state machine in which said

7   timer is deleted and said time-out message is queued to a non-existent state in a

8   synchronous state machine, wherein said handle function allows said application to

9   synchronously act on said timer.

1   Claim 33 (original): The method as recited in claim 31 further comprising the step

2   of:

3   stopping said timer by said application, wherein said timer is in an idle state in

4   said asynchronous state machine with said time-out message being queued, wherein

5   said time-out message is dequeued, wherein said handle function transfers said timer

6   from said idle state in said asynchronous state machine to an idle state in a

7   synchronous state machine, wherein said handle function allows said application to

8   synchronously act on said timer.

1   Claim 34 (original): The method as recited in claim 31, wherein said time-out

2   message is dequeued, wherein said handle function transfers said timer from said

14

3   running state in said asynchronous state machine to a running state in a synchronous

4   state machine, wherein said handle function allows said application to synchronously

5   act on said timer.